

51CTO.com  
技术成就梦想

我 们 只 谈 开 发

# 开发月刊

Development Monthly

2012年09月  
总第018期



编程赋予了我们什么样的改变

架构师成长历程：时刻对新技术保持敏感

Centos5.6下部署安装DRBD+Heartbeat+MySQL

developer.51cto.com



	<b>编程排行</b> <small>Billboard</small>
3	9月编程语言排行榜：Java并不应该被取代
	<b>专题报道</b> <small>《程序员的写实》</small>
6	编程赋予了我们什么样的改变
8	架构师成长历程：时刻对新技术保持敏感
10	Centos5.6下部署DRBD+Heartbeat
	<b>技术热点</b> <small>Techlogy hot</small>
11	2012年8月编程语言就业趋势
12	程序员为什么不使用Go语言呢？
13	深入了解MongoDB的Shell和文档
15	Windows RT应用程序开发介绍
17	即将成为程序员的你需要什么技能
19	第一门编程语言选谁？
21	程序员背着罪名还被当作是修电脑的
22	近期面试经历总结及公司评价（上）
24	近期面试经历总结及公司评价（下）
26	面试时，如何向公司提问？
27	一淘曲琳：搜索引擎的架构维护与职责协调
29	基于Java开发的开源QQ客户端iQQ
30	招聘优秀的开发人员：简历是很危险的
31	编程将死，程序员永存
32	美程序员12岁少年开发98款游戏
34	Facebook引发的HTML5危机
36	我在外包公司的一年测试生涯



# 9月编程语言排行榜

TIOBE 今天公布了九月份编程语言排行榜, 八月和七月相比来说, 并没太多显著的变化, PHP 从第 7 的位置上升到第六, 但比 2011 年来说仍有下滑的趋势。另外还有变化的则是 Lua 和上月相比上升了一名到达 17 的位置, 而 Ada 则下降至 18 名。

2012 年 9 月编程语言排行榜榜单

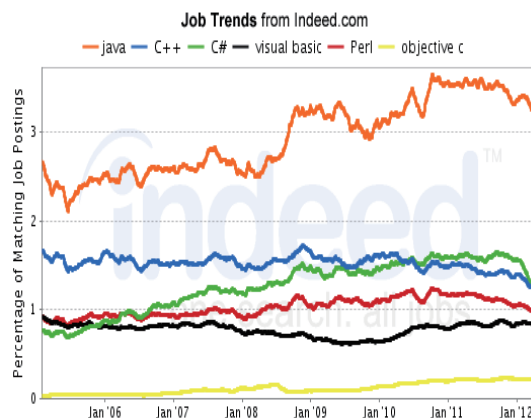
Position Sep 2012	Position Sep 2011	Delta in Position	Programming Language	Ratings Sep 2012	Delta Sep 2011	Status
1	2	↑	C	19.295%	+1.29%	A
2	1	↓	Java	16.267%	-2.49%	A
3	6	↑↑↑	Objective-C	9.770%	+3.61%	A
4	3	↓	C++	9.147%	+0.30%	A
5	4	↓	C#	6.596%	-0.22%	A
6	5	↓	PHP	5.614%	-0.98%	A
7	7	=	(Visual) Basic	5.528%	+1.11%	A
8	8	=	Python	3.861%	-0.14%	A
9	9	=	Perl	2.267%	-0.20%	A
10	11	↑	Ruby	1.724%	+0.29%	A
11	10	↓	JavaScript	1.328%	-0.14%	A
12	12	=	Delphi/Object Pascal	0.993%	-0.32%	A
13	14	↑	Lisp	0.969%	-0.07%	A
14	15	↑	Transact-SQL	0.875%	+0.02%	A
15	39	↑↑↑↑↑↑↑	Visual Basic .NET	0.840%	+0.53%	A
16	16	=	Pascal	0.830%	-0.02%	A
17	13	↓↓↓	Lua	0.723%	-0.43%	A-
18	18	=	Ada	0.700%	+0.02%	A--
19	17	↓↓	PL/SQL	0.604%	-0.12%	B
20	22	↑↑	MATLAB	0.563%	+0.02%	B

## Java 并不应该被取代

很多人都说 Java “已死” 这一类的话, 但是抱歉, Java 并不是你说一句就会被取代的。我们来看看 Java 开发人员的心声:《Java 已死? 九百万程序员说不》

Java 没死, 事实上它拥有足够的能量让你的应用跑起来。那些对 Java 吹毛求疵人频繁地聚焦在一些小众问题上, 总是和其他技术或者语言做些不公平的对比, 这些语言并没有像 Java 一样得到广泛应用及长远的历史。

## 再看 8 月编程语言就业趋势



从长期来看, C# 和 Java 都有强势的增长, 而其他语言在停止不前。虽然从短期来看 Java 呈现出了下滑的趋势, 但是仍然领先于其它一些编程语言。JAVA 语言作为云计算的主力支持语言呈现出来的简单、高性能、安全等特点将会更好完全发挥云计算高可靠、高拓展的优势。Android 是利用 Java 语言为基础进行开发的, 所以 Java 就业方向又多了一个, 就是 Android 手机开发。

## Java 是一门强类型静态语言

强类型静态语言有很多好处, 我喜欢它简单的视觉外观, 我可以粗略看一段代码就知道是做什么的, 它就像是可视化的用英语反馈, 语言非常易读, 字母混合成单词仍然可读。其它的一些好处是有强大的 IDE 支持, 动态语言在这点一直是

## 9 月编程语言排行榜 II

弊端,大项目中有强大的 IDE 和工具支持是无价的。

批判者的观点在 Java 在读文件、转换 xml 或迭代集合的时候缺乏表达力,但是你可以总是创建一个方法去处理这些常用的事例,或者是用 `FileUtils.readLines()`, java 语言上表达力的缺点有很多库对其支持。在 Java7 中能看到一些优雅的加强型功能,如自动关闭资源、`switch` 语句支持 `String`、数字类型支持下划线 (强烈建议读 Coin 项目)。Java8 承诺了更多的东西(最有意思的应该是闭包)。Java 是不是在任何方面都是那么的完美无缺呢?当然不是,这就是 Java8、Java9 在准备阶段的原因。我个人也不喜欢那些不太优雅的核心 API。是否的确将更多的事情留给平台而不是语言本身?java 核心包括 API 进阶设计跨度长达 20 年,API 的更新会破坏向后兼容性,有些设计太抽象,但有些还不够抽象,有些太零碎,有些根本不可思议。看看其竞争对手 .NET,核心 API 做的很好,比如统一的通信 API。Java8 在 Jigsaw 项目的帮助下,会有所改变。

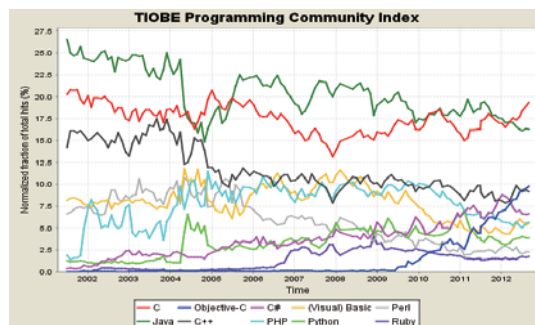
所以你值得拥有 Java,正确地使用它,它是一门很棒的语言。堪比 Klingon 语言,它将继续改善并不会很快消失。不应把努力放在取代 Java 上,而是和其他 JVM 语言结合使用,这是最明智的,但是对我的下一个 Pet Clinic 还是坚持使用 Java。

强类型语言 (静态类型语言) 是指需要进行变量 / 对象类型声明的语言,一般情况下需要编译执行。例如 C/C++/Java/C#

弱类型语言 (动态类型语言) 是指不需要进行变量 / 对象类型声明的语言,一般情况下不

需要编译 (但也有编译型的)。例如 PHP/ASP/Ruby/Python/Perl/ABAP/SQL/JavaScript/Unix Shell 等等。

### 前 10 名编程语言走势图



### 20 到 50 名语言排行

Position	Programming Language	Ratings
21	Bash	0.543%
22	Assembly	0.530%
23	SAS	0.528%
24	R	0.440%
25	COBOL	0.431%
26	Fortran	0.430%
27	ABAP	0.427%
28	RPG (OS/400)	0.410%
29	Scheme	0.401%
30	Logo	0.369%
31	Scratch	0.356%
32	Prolog	0.336%
33	D	0.311%
34	Haskell	0.303%
35	NXT-G	0.293%
36	Scala	0.273%
37	Erlang	0.266%
38	Smalltalk	0.249%
39	Common Lisp	0.241%
40	Forth	0.228%
41	APL	0.224%
42	ML	0.223%
43	Awk	0.220%
44	Processing	0.195%
45	ActionScript	0.187%
46	Alice	0.184%
47	C shell	0.163%
48	PL/I	0.162%
49	(Visual) FoxPro	0.155%
50	Tcl	0.154%

### 下面是第 50 到 100 的编程语言排名

ABC, Algol, bc, BETA, Boo, CFML, CHILL, CL (OS/400), Clean, Clojure, cT, Dart, Dylan, Eiffel, Emacs Lisp, F#, Factor, Gambas, Go, Groovy, Icon, IDL, Inform, Informix-4GL, J, JavaScript.NET, LabVIEW, Ladder Logic, LPC, MS-DOS batch, MUMPS, NATURAL, Oberon, OCaml, Occam, OpenCL, OpenEdge ABL, Oz, PowerShell, Q, REXX, S, S-PLUS, SPARK, VBScript, VHDL, WebDNA, X10, xBase, Z shell



Java 是由 Sun Microsystems 公司于 1995 年 5 月推出的 Java 程序设计语言(以下简称 Java 语言)和 Java 平台的总称。用 Java 实现的 HotJava 浏览器(支持 Java applet)显示了 Java 的魅力:跨平台、动态的 Web、Internet 计算。从此,Java 被广泛接受并推动了 Web 的迅速发展,常用的浏览器现在均支持 Java applet。另一方面,Java 技术也不断更新。

Java 平台由 Java 虚拟机(Java Virtual Machine,简称 JVM)和 Java 应用编程接口(Application Programming Interface,简称 API)构成。

Java 应用编程接口为 Java 应用提供了一个独立于操作系统的标准接口,可分为基本部分和扩展部分。

在硬件或操作系统平台上安装一个 Java 平台之后,Java 应用程序就可运行。现在 Java 平台已经嵌入了几乎所有的操作系统。这样 Java 程序可以只编译一次,就可以在各种系统中运行。Java 应用编程接口已经从 1.1x 版发展到 1.2 版。目前常用的 Java 平台基于 Java1.4,最近版本为 Java1.7。

Java 分为三个体系 JavaSE(Java2 Platform Standard Edition, java 平台标准版), JavaEE(Java 2 Platform,Enterprise Edition, java 平台企业版), JavaME(Java 2 Platform Micro Edition, java 平台微型版)。

Sun 公司对 Java 编程语言的解释是:Java 编程语言是个简单、面向对象、分布式、解释性、健壮、安全与系统无关、可移植、高性能、多线程和动态的语言。

51CTO 开发频道语

# 编程赋予了我们什么样的改变

人的生活、习惯、甚至爱好都会随着工作、环境、年龄慢慢开始偏移我们最初的生活轨迹。当这些凄凉而又在所难免的事情被我们拿出来互相分享交流时,也就意味着我们成熟了。

## 编程赋予了我们什么样的改变

人的生活、习惯、甚至爱好都会随着工作、环境、年龄慢慢开始偏移我们最初的生活轨迹。当这些凄凉而又在所难免的事情被我们拿出来互相分享交流时,也就意味着我们成熟了。当我们从大学校园走出来的那一刻,就已经注定了第一份工作和新的环境,将会使我们的生活发生天翻地覆的变化。

程序员这份工作在外行人眼里有多种代名词:“高手”、“黑客”、甚至“丝”;程序员的性格在外行人心里也有很多形容词:“木讷”、“古板”、“较真”……貌似“开朗”这个词跟程序员毫不相关;程序员的爱好在外行人眼中是:IT 新闻、IT 交流群、连发微博都是关于 IT 方面的,但是他们从没意识到那些他们认为流行的网站,其实是我们开发出来的。

为什么会产生这么大的误差呢?编程赋予了我们什么样的变化呢?下面听听两个工程师的心声,看的到底我们的生活轨迹偏离了多少。

## 满座网李工:学生时代的梦想走上的编程路

“自从做了这行以来,在他们眼里彻变成了一个宅男加工作狂。之所以被这么认为,还得从我喜欢上编程开始说起”,谈起编程工作带来的变化,李工回想起了当年选择这个行业的原因:“之所以选择了编程这一行业的原因就是因为以前上大学的时候看见了一个用 Java 编写的一个坦克

大战,虽然那类似打着马赛克并且只有几个简单操作的程序,还是给我带来的巨大的冲击力。所以就开始研究起了代码,通过学校一系列的学习,加上外面的辅导,我慢慢开始进入了程序员的这支队伍。之后便是我的第一份工作,是一家小型的互联网公司,虽然公司条件很一般,但对于当时的我来说却是很宝贵。那时负责的是后台开发工作,刚开始对于很多方面还不是很熟练,只能跟着团队一起走项目,从中获取些许经验,所以一有项目,加班就是在所难免的了,聚会等一系列社交活动只能取消了。记得有一年春节的一个项目,因为对方很对我们满意,领导一口承诺提前两个月上线,但是因为对项目的把握不够导致了偏差,这个项目还没有开始正式开发,结果我们为了这个不负责任的说话,只能天天加班赶这两个月,春节亦是如此。我想这也就是那些评价的由来了。”

当小编问起他上班前后的兴趣爱好时,这位工程师滔滔不绝的讲述了上班前很多的爱好,那种语气,就像是在怀念我们童年的开心事一样。他的兴趣跟大多数人一样,沉迷在球类和游戏的世界里。对比现在的爱好,有着质的变化,当他从一个菜鸟蜕变成一个有一些开发经验的程序员时,大部分休息时间都用在了博客这一块,经常会写一些自己工作上的感悟和迷惑,和网友沟通交流,对于别人对于自己都是一个成长的过程,这是一个很重要的过程,需要持之以恒,当你有所怠慢



## 编程赋予了我们什么样的改变 II

的时候,也许就离工程师远了一步。

成为工程师以后,负责的事情更多了,以前养成的一些习惯已经可以说完全不复存在,唯一保持的规律只剩下每天刷两次牙这一好习惯。以前从不爱好的游泳、健身等一系列运动,以后也更不可能出现。

在最后小编问他是否已经编出了自己心目中的那个坦克大战的时候,他笑了一下回答说:“坦克是写好了,就是还不能大战,因为没有时间。”

可以看出,也许在别人眼中他是个工作狂,但是在他自己心中,只是为了学生时代那个坦克大战的愿望雏形而努力着。

### 中法飞讯王工:玩游戏看热闹,写游戏看门道

王工在这家公司已经有一年零六个月的工作年头了。他的工作主要涉及一些 Java 后台、数据库方面的开发。平时用到的有 Struts2 Spring 跟 Hibernate 还有 Redis 跟 FreeMarker Quartz 等一些模版跟小框架。

在开发中也经常使用 Linux 进行开发。有时间的情况下也会研究下关于 Solr Sphinx 等搜索引擎的使用。

谈到对他个人的改变时,王工感觉最深刻的一点就是从原来的喜欢游戏,到现在的想了解游戏的一些底层实现,不再单单是去玩游戏,而更多的去琢磨游戏。并且看过一些关于游戏开发的完整教程,从最开始的场景拟定,到人物设定,和最后的剧情搭建,完整的看一遍,在动手实施一遍,改一些里面的角色,真是比玩游戏更有意思的多。从玩游戏到改游戏,王工完成了自己从一名玩家

到一名程序员的蜕变,而这样的蜕变对于王工来说,正是一个从青涩迈向成熟的过程,当然,这个过程源自编程的魅力。

从原来的喜欢看动漫,喜欢看电影,到现在喜欢看看业界的新闻和技术博客,觉得非常的有趣,而且又能开阔自己的视野并且对于工作起到了很大助力。

从以前爱围观路边争斗,到现在爱看 IT 业界的斗争,从 3Q 大战到现在的 3B 大战,其精彩程度不亚于新闻联播。

现在王工在朋友的眼中就是一个游戏研发中心,当他把他编出的一部分游戏展现给朋友时,朋友都会饶有兴致的玩上半天。最后还会问出一句让王工哑口无言的话:“你会写外挂吗?”。对于王工唯一没变爱好就是,以前喜欢坐着不喜欢运动,现在的工作让他依旧保持了这个以前的爱好。

### 编程到底赋予了我们什么样的变化?

从上面不难看出随着我们接触的层面不同,喜欢的事情也会变得跟以前有所偏差,这也许就是成长。有人说编程是一个头疼的工作,但是在有些人眼中。编程却是他们生活中不可缺少的乐趣。也有人说编程是令人作呕的事情,但是在有些人眼中编程却是令他们惊叹的艺术。是人生乐趣还是谋生手段?编程到底赋予了我们什么样的变化?这一切其实都由您来决定。

最后还要提醒所有程序员一句,适当早点洗睡吧,已经有程序员劳累过度而死了,无论是你的爱好还是工作,只有健康才能享受你的生活,就像一个爱好游戏的玩家,让他去日夜去做代练,也会演变成令人作呕的一件事。■

## ■ 编者按

架构师是一门建立在科学,工程基础之上的艺术,一个充满魅力,极其有趣的工作;给人带来了一种不被束缚,不按别人的思路走,自由描绘无限风景的期盼。那么程序员应如何成为一名合格的架构师?

# 架构师成长历程: 时刻对新技术保持敏感

架构师是一门建立在科学,工程基础之上的艺术,一个充满魅力,极其有趣的工作;给人带来了一种不被束缚,不按别人的思路走,自由描绘无限风景的期盼。那么程序员应如何成为一名合格的架构师? 如何加强自己的技能修养? 架构师在公司主要负责什么? 等等,记者带着网友的一系列问题专访了当当网技术部高级总监。曾历任开发工程师、高级工程师、架构师、高级总监。



傅强, 当当网技术部高级总监, 2006 年硕士毕业于北京航空航天大学计算机学院, 应届毕业生加入当当网。历任开发工程师、高级工程师、架构师、高级总监; 目前负责搜索、推荐与数据挖掘、个性化营销、广告联盟、用户行为分析、大数据技术解决方案。

以下回答基本以在当当网的工作为背景。

**51CTO:** 首先请您介绍一下自己的您的专长和目前的一些研究领域。

傅强: 06 年 -10 年, 专注于站内搜索。目前再次负责搜索这块业务;

07 年至今, 专注于通过数据挖掘、统计分析等手段, 开发 B2C 网站个性化推荐、个性化营销系统;

09 年至今, 专注于利用大数据处理能力, 解决用户行为分析;

11 年至今, 以用户行为分析为核心的 Anti-Fraud。

**51CTO:** 一开始您是因为什么开始对 IT 行业感兴趣的? 又是因为什么开始程序员的生活?

傅强: 我的本科、硕士都是计算机专业, 至少毕业时也没想到离开 IT 行业。

跟大多数程序员一样, 我在研究生期间最早开始程序员生活, 虽然是实习生; 毕业后若干年, 一直对编程保持激情; 至今仍然以“高级工程师”或“架构师”称号为荣。

**51CTO:** 对于任何一个软件开发人员来说, 架构师都是一个令人向往的角色。可以介绍一下您从程序员到架构师的过程中的一些经历吗?

傅强:

举个例子吧:

我在重构升级站内搜索服务时, 首先, 我对紧紧围绕公司搜索业务做了总结, 对 2 年内业务对搜索排序、性能等提出的种种需求做提炼—搜索大部分问题在于不同品类搜索排序效果相差较大, 对出现过的技术问题做总结—搜索的大部分



## 架构师成长历程：时刻对新技术保持敏感 II

投诉在于搜索结果中数据信息与网站单品详情页的信息不同步，即数据更新问题，对未来2-3年搜索业务的趋势做了分析—搜索流量支持10-20倍即可、搜索模型中点击负反馈信息量需要深度挖掘、大内存 SSD 磁盘的服务器越来越便宜等等，基于以上分析和总结，为新版搜索设计出更加合理的架构。

第二，对我本人之前几年在搜索方面积累的大量关键代码做重构，拆分若干功能独立的子模块作为搜索的基础类库。这让其他新加入搜索团队的成员上手快了很多，工作效率和产出提高很大。再加上我手把手细心地带新人，半年至1年，其中勤奋的程序员们的开发效率基本与我本人的开发效率相差不大了。这相当于复制出若干我自己。这样我自己不但带新人的能力有所提高，而且也在团队中建立了技术影响力。

第三，为搜索团队制定出代码规范、SVN 代码存储规范、测试流程规范、上线流程，并在坚持检查这些规范的执行程度。一段时间之后，团队在这些方面逐渐做的井井有条，从小作坊变成正规军。有了流程的保证，程序的质量也得到了保证。

以上是我从程序员成长为架构师时一次重要的关键点，供参考。

**51CTO：程序员要成为一名架构师发展，需要进一步加强技能的修养。在您的研究领域上，您是如何加强技能修养的？**

傅强：

几个必要的要素：

思考问题的角度尽量站的高一点、再高一点，从分解公司战略层面开始入手做架构设计

对业务的深刻理解，才能做好业务架构

理论学习 & 技术实践，什么时候都不能丢掉，这是技术架构的基础，重要性不赘述

同行业交流

时时对新技术保持敏感

**51CTO：个人觉得成为一名架构师仅仅通过理论学习是不行的，那么您的一个自我培养的过程是怎样的？也就是说从程序员到高级程序员再到设计师的整个过程如何去完成？**

傅强：

技术方面：

- 从最基础的开发做起
- 逐步提高解决高难度技术的能力
- 不断重构代码、不断优化代码，每次重构都是一次思考

业务方面：

- 从理解现有业务做起
- 从成为小领域业务专家，扩展到更多领域的业务专家
- 在每次重构底层代码时，更在不断思考业务架构重构与优化
- 做到以上，大约是个高级程序员或准架构师水平。

架构师的核心竞争力：

- 有独立思考、价值判定，建立在对行业、对公司战略和目标的深刻理解之上
- 很强的语言翻译能力，能跟产品经理讲明白技术，能把产品理念翻译成架构和可执行代码

■ 本文更详细内容，请查看：

<http://developer.51cto.com/art/201208/353684.htm>

## ■ 编者按

服务器平台是 Dell R410, 系统采用的 Centos5.6 x86\_64, 系统盘采用的是 Raid1, 然后单独拿了一块 SAS 500G 的硬盘出来准备给 DRBD 系统用。

## Centos5.6 下部署DRBD+Heartbeat

这里没有采用单独分区的做法, 这是因为在测试中我们发现, 单独分区做为 DRBD 系统使用时, 经常有如下报错(以下数据采集来自于实验环境)。特与大家分享。

报错信息如下所示:

Operation refused.

Command 'drbdmeta 0 v08 /dev/sda2 internal create-md' terminated with exit code 40

drbdadm create-md r0: exited with code 40

这个时候我们需要使用 dd 命令覆盖文件系统中的设备块信息, 如下所示:

```
dd if=/dev/zero of=/dev/sda2 bs=1M count=128
```

这个时候请一定要注意 dd 命令要清除的分区信息, 不要搞错了, 不然很容易将系统损坏, 我第一次实验时就遇到了这个问题; 如果部署 DRBD 时的分区信息是不需要写进 /etc/fstab 表的, 即不需要在安装系统时就挂载, 不然重启系统时很容易发生 Emergency 信息, 鉴于以上情况, 所以我这里建议大家还是用独立硬盘作为 DRBD 的设备。

两台机器的基本情况如下所示:

centos1.cn7788.com 192.168.11.32

centos2.cn7788.com 192.168.11.33

Heartbeat 的 vip 为 192.168.11.30

两台机器的 hosts 文件内容如下所示:

192.168.11.32 centos1.cn7788.com centos1

192.168.11.33 centos2.cn7788.com centos2

实验中暂时先用千 M 交换机的网络作为心跳线线路, 等实验搭建成功后再考虑加双绞线作为心跳线, 两台机器的 hostname 及 ntp 对时这些在实验前就应该配置好, iptables 和 SELinux 关闭, 具体情况略过。

### 一、DRBD 的部署安装

两台机器分别用如下命令来安装 drbd 软件, 如下所示:

```
yum -y install drbd83 kmod-drbd83
```

```
modprobe drbd
```

```
lsmod | grep drbd
```

正确显示如下类似信息, 表示 DRBD 已成功安装:

```
drbd          300440  4
```

两台机器的 drbd.conf 配置文件内容如下所示(两台机器的配置是一样的):

### 二、Heartbeat 的安装和部署

1. 两台机器上分别用 yum 来安装 heartbeat, 如下命令操作二次:

```
yum -y install heartbeat
```

如果你只操作一次, 你会惊奇的发现, heartbeat 第一次时并没有安装成功。■

本文未完, 完整部分请参考原文:

<http://database.51cto.com/art/201209/355332.htm>

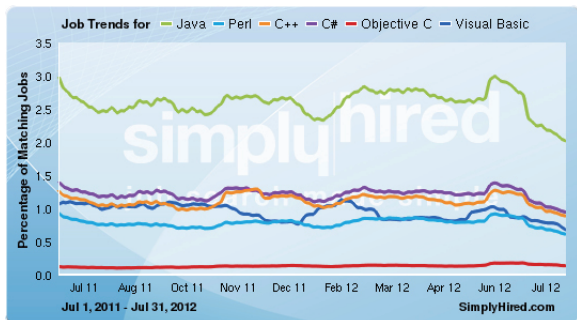
# 2012年8月编程语言就业趋势

这是整个夏季的就业趋势,看看你所喜欢的语言处于怎样的阶段。先来看看 Indeed.com 的就业趋势。

大部分语言的趋势都在下滑。OBJ-C 继续保持增长, C# 有显著的下滑,但仍然领先于 C++。从长期来看, C# 和 Java 都有强势的增长,而其他语言在停滞不前。目前的移动开发领域很红火,作为 iOS 的语言 OBJ-C 做了领头羊。C++ 和 Perl 有略微的下降,但并不是很明显。



再看看 SimplyHired 的短期走势

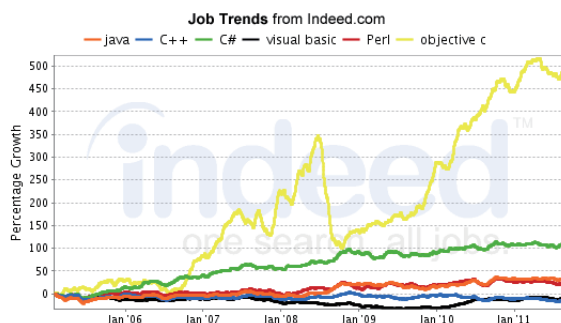


有趣的是,这里的 OBJ-C 并没有积极向上,但并其它语言的走势要好。Java 在近几个月出现了大幅的下滑,但仍然大幅领先于其他语言。C++ 和 C# 在过去的一年有几乎相同的趋势,在最近几个月都出现了下跌。

最后看下 Indeed.com 的相对图

Obj-C 增长最快,但已经放缓。C# 增长是稳定的,在过去 3 年徘徊在 100% 左右。Visual

Basic 和 C++ 继续下降。Perl 和 Java 是仍然增长的迹象,但不是很显著。



这意味着什么? 首先,移动开发非常火,但移动开发并没有积极影响 Java 的上升。为什么 Java 是相对增长,却不明显,主要是招聘帖子比以前少了,需求并没有从前那么旺盛。■



下载通道: <http://os.51cto.com/art/201208/355049.htm>

《Linux 运维趋势》是由 51CTO 系统频道策划、针对 Linux/Unix 系统运维人员制作的一份开放的电子杂志,内容从基础的技巧心得、实际操作案例到中、高端的运维技术趋势等等。■



## ■ 编者按

Go 是一个来源于 C 语言的编程语言,在保留了 C 语言的简约风格的同时也进行了很多改进。今年的早些时候我们对比过了几种 native 编程语言,包括 D 语言、Go、Rust 和 Vala。这里我们将分析来自 Google 的 Go 语言,看看他到底哪里跟别的语言不一样。

# 程序员为什么不使用Go语言呢?

Go 语言比其他的很多语言都吸引我,虽然我不是 Go 语言的专家,但是我很乐意在这里介绍一下它的特性,下面这些特性指引我转向 Go 语言。

## 快速简单的编译:

Go 语言编译的很快,事实上,他快的甚至可以作为脚本语言了。几个使他编译很快的原因有:

## 他不使用头文件

当 A 依赖 B, B 又依赖 C 时,那么首先会编译 C,然后是 B 和 A;但是如果 A 依赖 B,但是 A 并不直接依赖于 C,而是存在依赖传递,这时会把所有 B 需要从 C 拿到的信息放在 B 的对象代码里。这样,当编译 A 的时候,就不需要再管 C 了。

在编译程序时,只需将类型信息沿着依赖关系树向上遍历即可,如果到达树的顶端,则只需编译紧邻的依赖,而不用管其它层级的依赖了。

## 通过多返回值的错误处理:

现代的编程语言基本上有两种错误处理办法,例如在 C 语言里是使用返回值,而在 Java 等面向对象语言里使用异常处理返回值,因为返回值的状态码总是可能跟 需要返回的结果有冲突。Go 语言允许多返回值,从某种程度上解决了这个问题。你可以为你的函数的执行结果状态定义返回值,任何调用的时候都可以来检查,很方便。

## 简单的组合:

可以使用 interface 为对象指定一些类型的成员,还可以像 Java 一样给他们指定操作(行为)。例如在标准库的 io 包中定义了一个 Writer,就有一

个带有字节数组作为参数(输入)一个 integer 值和错误码作为返回值(输出)的方法。而实现了 io.Writer 接口中的 Write 方法的类型才是实际被执行的。这个设计能够非常优雅的分隔代码,还简化了单元测试过程,例如,如果你想测试一个数据库对象的一个方法,在传统的语言中你必须创建一个数据库对象,然后做很多协议初始化工作。在 Go 语言中,你可在接口下创建任何对象。

## 简单的并发:

在 Go 中并发变得非常的简单,在任何函数前方上 go 两个字母,这个函数就将以他自己的 go-routine (一个非常轻量级的线程)来运行,Go-routines 之间通过 channels 来通信。我们通常会有一些需要线程同步和互斥的需求,在 Go 中非常简单,Go 只是启动并发任务,各个任务之间通过 channels 来通信,从而协调同步和互斥。

## 优秀的错误提示:

我从没见过别的语言有 Go 语言这么高的错误诊断质量。例如如果你的程序思索了,Go 的运行时可以通知你,而且,他甚至可以告诉你哪个线程出了问题。当然编译错误也是很详细很有用的。

其他特性:Go 语言还有其他非常吸引人的特性:高阶函数、垃圾回收、哈希映射、可扩展的数组等等。■

# 深入了解 MongoDB 的 Shell 和文档

■ 编者按  
MongoDB 其实是数据库 shell. 一般假定它和 mongod 运行在同一台机器上, 还假定了 mongod 绑定了默认端口。.....

## 1.MongoDB shell

MongoDB 其实是数据库 shell. 一般假定它和 mongod 运行在同一台机器上, 还假定了 mongod 绑定了默认端口。

如果不是这样的话, 可以在启动时指定这些参数, 让 shell 连接另一台服务器:

```
mongo 192.168.10.10:10000
```

这样就会连接运行在 192.168.10.10 上端口为 10000 的 mongod

shell 默认连接 test 数据库, 要使用别的数据库, 在服务器地址后添加斜杠和数据库名:

```
mongo 192.168.10.10:10000/  
refactor
```

这样会连接 192.168.10.10:10000 上的 refactor 数据库

也可以用 --nodb 选项启动 shell, 而不连接任何数据库, 而只是试试 javascript

```
mongo --nodb
```

db 绝不是仅有的数据库, 从 shell 中可以连接任意多的数据库, 这对多个服务器的环境很方便. 调用 connect(),

并将结果赋值给变量. 如在分片环境中, 可能想用 mongos 表示 mongos 服务器, 还想要有每个片的连接, 可以

如下操作:

```
mongos=connect("127.0.0.1:10000"
```

```
)
```

```
shard0=connect("127.0.0.1:10001")
```

```
shard1=connect("127.0.0.1:10002")
```

这样就能将 mongos, shard0, shard1 作为 db 变量使用.

## shell 工具

对于管理多个数据库, 有多个数据库变量就比简单的 db 有用处, 如在分片中,

要维护一个单独的只想配置服务器的变量:

```
config=db.getSisterDB("config")
```

```
config.shards.find()
```

## 2.BSON

MongoDB 的文档是个抽象概念, 其具体的呈现形式取决与使用的驱动和编程语言.

因为 MongoDB 中的通信大量依赖于文档, 所以需要一种所有驱动, 工具和进程都能共享的文档表达方式.

这种表达方式叫 Binary JSON(BSON)

BSON 是轻量级的二进制格式, 能将 MongoDB 的所有文档表示为自己字符串. 数据库能理解 BSON, 存在磁盘上的文档。

也是这种格式.

当驱动要插入文档, 或者将文档作为查询条件, 驱动会将文档转化成 BSON, 然后再发往服务器. 同样, 返回到客户端的文档。

## 深入了解 MongoDB 的 Shell 和文档 II

也是 BSON 格式的字符串。驱动需要将这些数据解码,编程本机的文档表示,最后返回给客户端。

用 BSON 的三个主要原因:

### a. 效率

BSON 设计用来更有效的表示数据,暂用更好的空间。最差的情况下,BSON 比 JSON 效率略低,最好的情况下(

比如存放二进制数据或着大多数),BSON 要比 JSON 要高效的多。

### b. 可遍历行

有些时候 BSON 牺牲了空间效率,换取更容易遍历的格式。如,在字符串前面加入其长度,而不是在结尾处使用一个终结符

这对 MongoDB 的内嵌文档很有用。

### c. 性能

BSON 编码和解码速度都很快。它用 C 风格的表现方式表示类型,在大多数编程语言中都很快。

## 2. MongoDB 的传输协议

驱动在 TCP/IP 协议的基础上简单封装了 MongoDB 传输协议,用来与 MongoDB 交互。

MongoDB 的传输协议基本上是对一个简单封装的 BSON 数据,如,插入消息会有 20 字节的头部数据(包括

告知服务器执行写入操作的代码,以及消息长度,要插入的集合名,要插入的 BSON 文档列表)

## 3. 数据文件

MongoDB 的数据目录中,每个数据库都有几个独立的文件。每个数据有一个 .ns 文件和若

干数据文件,数据文件以

递增的数字结尾,所以,数据库 refactor 会被存放在 refactor.ns,refactor.0,refactor.1 等文件中。

每个新的以数字结尾的数据文件大小会加倍,直到达到最大值 2GB。这是为了让小数据库不浪费太多的磁盘空间,

同时让大数据库使用磁盘上连续的空间。

MongoDB 为了保证性能还会预分配数据文件(可以用 --norealloc 关闭这一功能)。预分配在后台完成,有数据文件

被填满时就会自动启动。这意味着 MongoDB 服务器总是视图在每一个数据库保留一个额外的空数据文件来避免

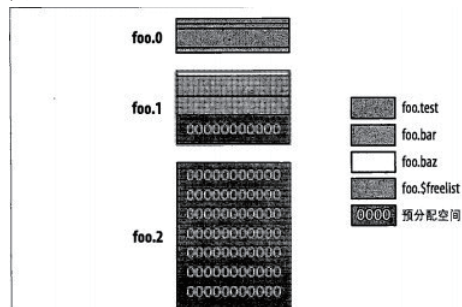
文件分配产生的阻塞。

## 4. 命名空间和数据域

在数据文件内部,每个数据库都是按照命名空间组织的,一种类别的数据与其它类别的分开存放。每个集合的文档都有自己的命名空间,索引也是。命名空间的元数据存放在数据库的 .ns 文件中。

每个命名空间的数据都被分成若干组,放到数据文件的某一个区域内,这个区域成为数据域。

如图:



数据库 foo 有 3 个数据文件,其中第 3 个是预分配的空文件。■原文未完,更多请查看:

<http://database.51cto.com/art/201208/352894.htm>



# Windows RT应用程序开发介绍

## 1 概述

这篇的标题叫做 Windows RT Introduction 而非 Windows 8 Introduction 是想强调此次介绍是从开发人员的角度而不是普通用户的角度出发的。同时,我们关注的是 Metro Style 应用而不是传统的 Win32 应用程序的开发。

实际上,使用 C# 或者 HTML + Javascript 书写一个 Hello world 应用的代码例子已经在网上泛滥了。但是仅有一个 Hello world 并不能够说你掌握了 Win RT 的开发。从 Pro 的角度来说我们应该弄清楚整件事情的细节。那么首先就应当是他的架构。这样写起程序来才能心定。

## 2 Windows 8 Metro 与 Desktop 模式

### 2.1 两种模式

Windows 8 的应用程序显示模式目前有两种,定义在 METRO\_MONITOR\_MODE 中:即传统的桌面模式(MMM\_Desktop)以及 Metro 模式(MMM\_Metro)。如果你是 Windows Phone 用户的话可能会对 Metro 比较熟悉。事实上,微软在 2009 年启动 Windows 8 的研发工作时目标是创造一个完全不同以往的操作系统,完全不以之前的操作系统为蓝本。而后发现 Desktop 应用是不可或缺的部分而将两个部分进行合并。一开始用可能会有些别扭,但是我估计开发人员半天之内就能够熟练使用这个系统了。

### 2.2 Metro 和 Desktop 的一些不同

既然有两种模式那么我们自然就会关注他们的不同点。这个问题应该从架构图上做一下说

明但是我们可以先有一些直观的认识。

#### 2.2.1 Message Loop

消息处理的编程是传统 Desktop 应用程序的重要部分。你需要书写维护 Message Loop 的代码。例如:在 WinMain 调用(或者其子例程中)你需要书写类似

```
while (::GetMessage(&message, NULL, 0, 0)) {  
    ::TranslateMessage(&message);  
    ::DispatchMessage(&message);  
}  
  
而在 Window 创建之前你一定指定了  
WNDCLASS wndClass;  
  
// ...
```

```
wndClass.lpfnWndProc = WndProc;
```

这样你就可以在 WndProc 函数中决定特定 message 的流向了。对于绘图来说,你一定是接受了 WM\_PAINT 消息,然后执行了区域重绘。

但在 Metro App 中这些都已经隐藏了,而且消息的细节也可能发生了变化。Metro App 中你看不到消息循环。一切关于界面消息的分发都隐藏在了 CoreDispatcher 中。因此如果你用 Spy++ 去试探 Metro App 的消息循环那么你什么都抓不到。

#### 2.2.2 Display

在传统的 Desktop 应用程序中,绘图可能通过 GDI, GDI+, DirectDraw, DirectX 进行。同样通过捕获 WM\_PAINT 消息或者当系统处于 IDLE 的时候进行绘图(对于游戏编程来说)。

而 Metro App 不会再支持 GDI 和 GDI+,

## Windows RT 应用程序开发介绍 II

在 Metro App 中绘图只能通过 DirectX 来进行。确切的说是 Direct3D 和新公布的 Direct2D、Direct Write API。因此 Metro 应用的所有绘图都是希望是硬件加速的。这种绘图更高效,解放 CPU,而且一般不需要处理复杂的 Dirty Region Repaint。

### 2.2.4 Share & Communication

传统的桌面应用程序有多种手段进行公共组建的公用或 IPC。但是在 Metro App 中,隔离是一个很重要的概念,应用的可执行部分,运行库, Isolated Storage 都是独立的,不能够共用。同样,不能够使用传统的 IPC 机制。应用程序的互动仅仅可以通过内置的 Contracts 进行,关于这一部分内容可以查看 MSDN:

<http://msdn.microsoft.com/en-us/library/windows/apps/hh464906.aspx>

### 2.2.5 Portability

传统的 Desktop 应用程序的支持大多为 x86/64 架构的处理器。由于 Metro 环境可以完整运行在 ARM 处理器上是一个重要的特性,因此 Metro App 可以运行在 ARM 处理器上,即同时部署在 PC 和移动设备上。

### 2.2.6 OS Access

当然为了 Portability 的要求,必然要求应用不能够越过 Win RT 的抽象,因此 Metro 是不能像 Desktop App 那样访问所有的 Windows API 的。

3 从 Windows 8 API 的架构图看 Windows RT

我们对 Windows RT 的介绍都将围绕着这个图展开。

在这个图中,最底层的是 NT 的内核;在其上是 Windows 子系统。实际上 NT 至少有三个子系统,Windows 子系统,POSIX 子系统(Unix)和 OS/2 子系统。POSIX 子系统和 OS/2 子系统实际还是在使用 Windows 子系统。在 Windows 子系统上划分了不同的运行时(橙色)和程序库(浅蓝色),最上面的绿色是我们使用的各种开发语言。

这个架构图实际上说明了一切。并且消除了很多误解:

(1) 第一个误解是 INFOQ 指出的 Windows RT 和 Win32 是完全分开的。这源于微软发布的一幅饱受批评的架构图,在那张架构图中,Windows RT 和 Windows 子系统竟然是并排排列的。这是很荒谬的,Windows RT 实际上基于 Windows 子系统。首先 Windows RT 完全基于 COM;其次 Windows RT 利用了一部分现有的 Win32 API;其余的部分 Windows RT 则直接访问 NT 内核。

(2) 第二个误解是 C++/CX。C++/CX 是微软推荐的开发 Windows RT 的方式。他主要隐藏了 COM 的复杂性。关于这个问题我们后续会有说明。这个误解是 C++/CX 实际就是 C++ CLI。实际上这是两个完全不同的东西,C++ CLI 是运行在托管环境下的,而 C++/CX 完全是 Native 的。

### 3.1 Windows RT 仅用于 Metro 应用

从架构图中可以看出,Win RT 仅仅用于 Metro 应用。■本文未完,请参考原文:

<http://developer.51cto.com/art/201209/355507.htm>

# 即将成为程序员的你需要什么技能

编程语言不下百种,例如 C、C++、Java、JSP、SQL、PHP、JavaScript 等等等等,还有很多没有列举出来的语言。对于要开始找工作的大学生可能会有一个想法:‘我要是找一份工作,都需要学到什么语言’。

## 即将成为程序员的你需要什么技能

九月,一个新学年的开始,有满怀憧憬的大一新生报到入学,也有即将进入社会实习的大三、大四学生们,上大学时老师可能会想法设法激起你学习的兴趣,甚至说哄着你学都不为过,但是进入社会后,你只能靠自己,没有哪个老板会哄着你让你好好工作。

即将进入程序员这支队伍的大学生不在少数,你是否在即将进入社会之前心中产生种种不安呢?你是否还觉得自己的技术不过硬呢?其实有一些不会是很正常的,毕竟对于刚进入工作的大学生很多公司会采用慢慢培养的方式,但是你不能有两项不会,那就是这些也不会,那些也不会。

编程语言不下百种,例如 C、C++、Java、JSP、SQL、PHP、JavaScript 等等等等,还有很多没有列举出来的语言。对于要开始找工作的大学生可能会有一个想法:‘我要是找一份工作,都需要学到什么语言’。现在小编就来带大家了解即将成为程序员的你需要什么技能。

### Python 工程师

我们来听一位正在国外做科研的博士 Vamei (网名) 个人看法:‘如果针对 Web 开发的 Python 工程师,个人觉得需要基本懂得:Python + Django/Twisted/Web.py 等。Python 包括基本的

Python 语法和标准库。后面是常用的 Python 做 Web 开发的框架。这样,就可以足以布局一个成熟的网站的后端,个人觉得也是 Python 现在应用最广泛的地方,所以找个工作应该不难’。

但是 Python 想要掌握好的话,需要学习一些系统和网络的知识。

操作系统基础: 文件系统,进程,线程,信号。

网络基础: TCP/IP 协议, HTTP 协议, 一些 Web 前端的知识 (HTTP+CSS+JavaScript+jQuery)。

Vamei 认为: ‘Python 是一个很好的工具, 作为动态语言, 可以很方便地写出可用的程序, 来管理文件和服务。此外, 还会需要用到 C 和 Fortran, 作为模型的开发工具。这两者的性能都比 Python 高。Fortran 的话, 因为科研方面应用比较多, 所以还不得不依赖。最后是 Matlab, 很常用的数据处理和绘图工具。R language 用于统计的语言。’ 最后你可以进入他的博客学习到更多关于 Python 的知识。

(vamei 的博客: <http://home.cnblogs.com/u/vamei/>)

### Java 工程师

一位优秀的美女软件开发工程师王女士的看法: ‘作为刚毕业的大学生, 你的基础是最重要的, 基础一定要好。你还要了解 Java 的各种语法知识, 当你的编辑器报错的时候, 你必须知道是



## 即将成为程序员的你需要什么技能 II

什么样的语法错误,并且能够进行修改。还需要知道 J2EE 的各种知识,至少要了解 SSH 框架,并且其他常用框架要是会则更好,如: Spring MVC、iBATIS,当然,这些并不太过于奢求。还有就是 API 这个核心部分也要了解,你若是不熟悉 API,那么你不可能熟练运用 Java。最后基本的数据库的增删改查你要牢记在心。’

‘当然各个公司要求的是不一样的,你必须针对分配的项目来学习需要的知识。例如我的第一个项目用到的则是 ZK 框架,对于国内并不流行的 ZK 框架来说,连搜索引擎都不能给我满意的答案。但是要记住,不要抱怨,你的抱怨换不来任何好处。别看招聘信息上写的非常的吓人,称之为残暴都不为过,但是请别灰心,你若是照着大部分招聘信息上的要求来修改简历,等你面试的时候,项目经理或技术出身的大牛看了会笑着说:你这能力是我们公司最厉害的工程师才能达到的。’

在往后的深入学习你需要掌握软件架构设计思想、搜索引擎优化、缓存系统设计、网站负载均衡、系统性能调优等实用技术。当然,这些都是很深很复杂的东西,你刚开始并不需要掌握,也没有人会要求一个刚毕业的大学生掌握这些,但是你若是想要更好的发展,那么请开始慢慢学吧,也许是五年,也许更久。

### PHP 工程师

PHP 软件开发工程师 WX (网名)的网络看法:‘我两年前开始找工作的时候人家面试就分为两块,一是 PHP 基础,常用的命令及选项,另一方面就是 MySQL 的知识。这两方面是极为重要

的,其次还要求 HTML+CSS+JavaScript+Ajax 这些能看懂就好。当然,刚开始不让你会用,但是在以后的工作中你一定要学会如何去用,不然你自己也不会好意思老去求别人。’

若是想升职的话,你就要把 PHP 主流产品玩熟,也就是框架结构。例如 Discuz/ShopEx/ECShop/ThinkPHP。这些都是成熟的产品,而你需要的则是读懂他的处理方式。当你都知道了以后你基本就可以开始带项目了。再高深一点,就要涉及到架构的设计思想,还有针对搜索引擎的优化和网站的负载均衡等。

### .NET 工程师



(本文未完,请阅读原文) ■

<http://developer.51cto.com/art/201209/355425.htm>

# 第一门编程语言选谁？

这篇文章是专门针对大学低年级学生(和其他软件开发初学者)写的,如果你已经是研究生或本科高年级学生,请将这篇文章转发给你的师弟或师妹,希望这篇文章能够帮助他们少走弯路,顺利地迈入软件开发的大门。

## 说明:

这篇文章是专门针对大学低年级学生(和其他软件开发初学者)写的,如果你已经是研究生或本科高年级学生,请将这篇文章转发给你的师弟或师妹,希望这篇文章能够帮助他们少走弯路,顺利地迈入软件开发的大门;如果您是一位有经验的软件开发者,或者是关注计算机教育的同行,也敬请提出宝贵意见。

最近,台湾知名技术专家蔡学镛先生写了一本《编程 ING》,宣称“人人都能学会程序设计”。作为一名 IT 教育工作者,这本书引发了我的兴趣,翻看之后,共鸣之处不少,结合国内计算机教育的现状,产生了颇多感触,于是就有了这篇小文。

## 一、为什么学生视编程为畏途？

先当学生后当老师,不知不觉之中我在大学里已“混”了十多年,我发现,进入计算机专业就读的学生,最初至少有一大半对真实的软件开发根本不了解,是“一张白纸”,不幸的是,学了四年之后,许多张“白纸”又变成了许多罐“浆糊”,带着对软件开发可能是畏惧也可能是无所谓但绝对不是喜欢的感触离开校园。

编程真的那么没劲? 那么难和枯燥?

我写了将近二十年的代码,虽然不靠编程吃

饭,但也似乎勉强可算是个老程序员,我对编程的看法可总结为两句:何以解忧,唯有编程! 我经常在想一个问题:编程其实是很有趣很好玩很实用并很有成就感的一件事,为什么会有这么多的学生视编程为畏途? 而我们的计算机教育,为什么在打掉学生对编程的兴趣方面“如此成功”?

蔡学镛先生在《编程 ING》给出了一张图:

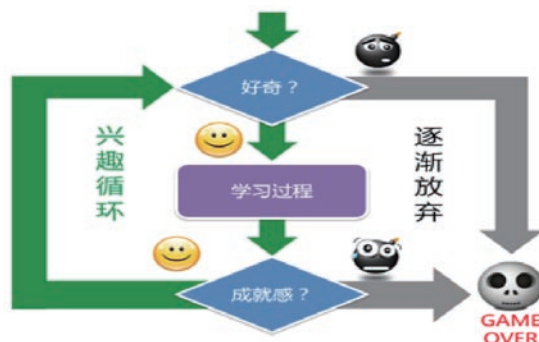


图 1 正向兴趣循环是学习的关键

我认为这张图道出了问题的关键——学习过程中的“正向”兴趣循环是否成功地建立。

强烈的兴趣与不断获得的成就是整个学习过程的“引擎”,它为学生完成整个学习任务提供源源不断的强大动力。有无数的事实支持这个观点。

传统的教学观点认为,本科的主要教育目标之一是为学生在本专业领域未来的发展“打下扎实的理论与实践基础”,所以从一开始就要“严格要求”,“科学训练”。

## 第一门编程语言选谁？ II

这个观点不能说错,但我认为,我们的计算机教育,尤其是针对初学者的教育,首要的任务是引发兴趣。没有兴趣,一切免谈。

我所了解的事实是:计算机专业的学生有不少视编程为畏途。其原因在于我们的现有计算机教学方式从一开始就给了这些学生“痛苦”的编程体验,不幸的是,这种体验在后期枯燥的专业课学习中不断得到强化,学生最终对编程敬而远之或畏之如虎。

事实上,教育学研究早已指出,成功的高效的教学应该是这样的:循序渐进,由浅入深,步步为营,兴趣导向。

教师的职责,不是将知识“灌入”学生的大脑,首要的任务是引发学生的兴趣,鼓励他们去探索未知的领域,主动地学习和吸收知识,培养技能,积累经验。在这个学习过程中,教师要成为一名优秀的导航员,给学生绘出航线,鼓励他们出海远航,解决他们在航行中所遇到的困难,并帮助学生建立学习的“正向”兴趣循环。

对编程的“第一印象”很重要啊!由此,引发了一个很有趣的问题——应该选择哪一门语言作为学生的第一门编程语言?

### 二、你学的第一门编程语言是什么?

在国内的大学中,当前大多数选用C作为学生的第一门编程语言。这其实并没有太大的问题,C的重要性无须我多说。其实问题的关键不在于选择C教学,而在于以哪种方式去教。

很不幸,国内许多C语言的教材都将主要的精力放在对C语法细节的介绍上,课程考核方式

又很古板——很多院校采用闭卷考试,出一堆的选择题和填空题。典型的题目是将一段代码砍掉一两句,让学生“填空”。有哪位高手是通过做这些“填空题”学会编程的?上机也流于形式,让学生反复折腾几个“黑底白字”的“玩具般的”小程序,学了一个学期,学生连一个有点用的程序都写不出来……

这种僵化的教学方式,足以毁掉多数学生对编程的兴趣。

我个人认为,C不应该成为针对大多数学生所讲授的第一门编程语言,我们的教学体系,应该给学生提供更多的选择。

针对初学者所讲授的第一门编程语言,应该具有以下的特点:

(1)必须是“有趣”的,能诱导人去“动手”和“思考”。

(2)需要对初学者屏蔽不必要的底层技术细节,以免分散他们的注意力。

(3)这种语言必须足够简单,但同时又具备足够的能力编写出实用的程序,从而让学生能比较容易地获得成就感,感悟到软件开发的魅力。

(4)这种语言必须能充分地体现现代软件开发的基本思想和技术成果,为学生进一步深入学习打下基础

(5)花在这门编程语言上的时间和精力是有回报的,掌握了它,就掌握了一个强大的工具,可以在今后的学习中使用这个工具进行实践和创造。(本文未完,请阅读原文)■

<http://developer.51cto.com/art/201209/355214.htm>



# 程序员背着罪名还被当作是修电脑的

家里最近装修,我就在家里布置网线,以前都不会剪水晶头,不得不从头学起。

家里是电信宽带路由器共享上网.三台电脑,路由器自动拨号,然后开启DHCP服务就可以了。

但是最近才发现一个比较严重的问题:两台台式机都可以上网,但是我的笔记本就是上不了网.感觉笔记本被歧视了,

没办法,有问题还是要解决的.笔记本总是提示本地连接连不上,网络设备连接出现错误,而且是断断续续,不停的连上,网络设备无法连接,就这样子不停的提示.路由器端口上面的信号灯显示断断续续,没有其他端口那样子一直亮着.笔记本不管是XP还是Ubuntu都是这样子的,当然Ubuntu是看不到提示的,但是连不上网还是很容易发现的。

刚开始有点抽筋了,以为是自己的网卡坏掉了,因为台式机都可以正常上网.但是后来发现问题不是那么简单.当我拔掉其他两台台式机的网线时,笔记本就可以正常上网了.于是我就开始检查笔记本的网卡设置,后来找到了解决的办法。

先说window吧,我先是尝试配置网卡电源选项,去掉"允许计算机关闭这个设备以节约电源"选项,但是还是不行。

然后我就配置网卡属性,发现网卡连接速度默认设置是"自动",刚开始没感觉这地方会有问题,家里的宽带是2M,自动检测应该会先选择10M速度吧~,但是问题恰恰就在这里,不过后

来还是发现了.我发现台式机都是默认10M全双工模式,所以果断将笔记本网卡设置为10M全双工模式.XP不能上网的问题就解决了。

再说说ubuntu吧,已经找到了问题,就是在ubuntu下设置一下网卡,可以使用mii-tool.现将网络连接断开,然后运行命令sudo mii-tool -F 10baseT-FD eth0,然后再运行sudo mii-tool就可以看到网卡已经是10M全双工模式了.然后连接网络就可以上网了。

这里有点想吐槽的东西,回家半个月,几乎隔天就要修电脑.不禁自问:计算机专业 == 修电脑?我可以抱怨吗?

**以前也是经常有些同学聊天就问:你会修电脑不? 你会盗帐号不?**

第一,会不会修电脑?这个问题我想,我可以回答:我不会.学计算机专业什么时候和电脑修理工挂钩了?在大学里面,电脑维修,很多时候,不错都是计算机专业的同学在做,但是那些电脑问题都只是小问题,不涉及到硬件,最坏的结果就是重新装系统.不知道什么时候,我变成电脑修理工了.回家,亲戚都让我去给他们修电脑,帮忙我当然是很乐意的,也都不是什么大问题.但是就是心里感觉很不爽.尤其是当你遇到一些人,很理直气壮的就认为你是计算机专业的,理所应当就会修电脑.此刻,真的是中枪了~

第二,会不会盗帐号?这个问题,我只能回答:很高深.(本文未完,请阅读原文)■

<http://developer.51cto.com/art/201208/354102.htm>

# 近期面试经历总结及公司评价（上）

## 前言

之前在博客园看到有朋友发表过一些工作经历以及面试体验,由于本人近期也在考虑换工作的问题,一个月来也陆陆续续面试了一些公司,下面就把我的面试经历以及个人对公司的印象同大家分享一下,希望对大家有所帮助。首先需要声明的是本人 3 年 JAVA 工作经验,绝对的凡人一枚,不像某些人那么牛 X,可以去 Google、微软、雅虎等这些顶级公司面试。所以大家看了觉得不好的话勿喷,如果觉得还不错,有所帮助的话麻烦顶一下。谢谢!

## 中海纪元

这家公司是本人面试的第一家公司,地址在魏公村附近。由于去的时候已经是中午 11 点多了,公司并没有安排笔试,直接就带着简历进入会议室面试了,面试的时候是两个人,一个人力,另一个是技术。人力的小 MM 上来就先发制人,问了一个哥们我特别蛋疼的问题:“请问你从毕业到现在一共写过多少行代码?”;本人当时就被这个问题所深深的折服,接着又陆陆续续问了几个相当业余的问题后,自信满满的离开了会议室。剩下的就是我跟技术的面谈,大概聊了有将近半个小时的时间,主要问题就是围绕着 Hibernate、mybatis、spring、struts 这几个框架所展开,期间也简单说了下 HTTP 协议、session、单元测试等问题,总体来说面试的难度相对简单。之后那个技术就又把人力小 MM 叫进来跟我谈薪水,由于本人要求的薪水,该公司给不到,所以后来也就不了之。

面试难度: ★

评价:问题主要还是基于 SSH 的使用以及配置,并没有太深过入。

工作环境: ★★

评价:公司的办公环境一般,而且面积略显狭窄。

## 五八同城

在面试的这一个月中,单就工作环境来讲,五八同城应该是给我印象最好的一家公司,公司充分体现了人性化的一面,公司有四层,每一次都有不同的娱乐休闲设施,比如:按摩椅、桌上足球、围棋、象棋、跳棋等,而且下午还会有免费的水果。五八没有笔试、直接就是面试,面试时只要把自己的能力尽量的全部展现出来,让面试官对你有个大概的了解之后,会问一些跟你项目经验有关的问题,但是不会很难,之后基本就是跟人力谈薪水的问题了。其实本人当时差一点就选择了这家公司,但是中间出了一些小状况(这里就不细说了,只能说相当离奇,估计这辈子都不会出现第二次),最后还是选择放弃。或许是天意如此吧,时也命也!不过需要说的是 HR 相当坑爹,最好跟 HR 谈的时候能当天决定的就当天决定,一定不要等他给你打电话!因为你是等不到的。。。

面试难度: ★★★

评价:对于有工作经验、能力还不错的人来说,面试的难度不是特别大,只要尽力展示出自己的长处就 OK。

## 近期面试经历总结及公司评价(上) II

工作环境: ★★★★★

评价: 公司面积比较大, 感觉很宽敞, 而且每一层都会有不同的娱乐休息设施, 给人感觉比较舒服、贴心, 在工作之余还可以放松心情。

### 易宝支付

到易宝支付面试的时候是在 12 点 40 左右, 由于正处于休息时间, 前台就让我坐在沙发上等到 1 点。期间感觉公司的环境还不错, 相当宽敞, 而且有种绿色生活的惬意。等到 1 点的时候, HR 过来说你走错楼层了, 技术部在 10 层。结果跑的 10 层发现, 坑爹啊!!! 这工作环境落差也太大了吧, 灯光灰暗, 气氛相当压抑, 每个办公桌上的人都表情严峻的忙碌敲打着键盘, 仿佛有做不完的事情。当时给我的第一感觉就是“我想马上离开”, 但是既然大老远的跑了过来, 不能白跑一趟吧。首先做了笔试题, 笔试题难度适中, 主要是一些 JAVA 基础、以及数据库表设计、智力测试题, 做完之后, 就跟技术到了一个空闲的地方开始聊, 一开始当然是千篇一律的自我介绍, 自我介绍之后, 本人这次决定反客为主, 上来就说起了 tomcat 源码以及 tomcat 中 session 的实现与管理。之后面试官觉得不错就陆陆续续问了一些 JAVA 中异常的相关知识以及 HTTP 协议、JAVA IO、多线程、设计模式、线程池的原理、SSH、ArrayList、HashMap 等容器的实现原理。由于问的比较多、而且基本上覆盖了 Java 大部分的知识点, 这里就不细说。大部分都回答上来了, 只是有些深入的问题自己确实不清楚。但是由于总体表现不错, 所以技术这关就顺利通过了, 二面是技术总监的

面试, 主要问的就是自己以后的规划、以及工作中碰到哪些问题、如何解决之类的。之后让我回去等通知, 第二天 HR 就打电话让我去复试, 由于考虑到工作环境以及路程的问题, 就回绝了。当时 HR 还争取了一下, 但是本人对那样的工作环境实在无法接受。不过从 HR 口中了解到原来面试的是核心平台的 JAVA 开发, 那就难怪面试相对来说比较难了。

面试难度: ★★★★★

评价: 问题涉及比较广, 而且有一定难度, 要求应聘者 JAVA 基础相对扎实, 主要问题还是集中于 JAVA 基础方面, 而且会不断深入, 并要求给出解决方法。

工作环境: ★

评价: 工作环境真的不敢恭维, 灯光暗的就跟阴天一样, 给人一种不舒服的感觉, 而且座位之间间隙比较小, 显得人多又拥挤。

### 用友致远

用友致远是用友软件的一家子公司, 公司位置相对偏僻, 本来约的是 11 点, 结果当我找到那里的时候已经是中午 12 点了, 人家已经到了吃饭时间, 所幸旁边有个篮球场, 就在那边跟他们公司的人干了一场, 战况十分焦灼, 最后我们临时拼凑的四人组还是以两球之差惜败。结果到了下午面试的时候已经处于疲惫不堪的状态, 因为下午公司停电, 所以组织出游了, 公司没什么人, 好在面试官还在!

(本文未完, 请阅读原文) ■

<http://developer.51cto.com/art/201208/354695.htm>

# 近期面试经历总结及公司评价（下）

## 搜房网

这家公司位于西直门外大街西环广场,第一次到那边,感觉人流量非常大,而且附近就是商业区。到了公司之后,面试官将我带到一个会议室,进行简单的面试,首先是自我介绍,然后叙述一下项目经历,当项目经历介绍到一半时,被面试官打断,要求做一下上机测试。机试题目并不是很难,大概就 5 道题,2 道 CSS 样式题、2 道脚本题(但要求分别使用 js 跟 jquery 实现),还有 1 道是 java 代码纠错。除了 css 样式实在不怎么会,其他的都写了出来。当我写完通知面试官后,面试官叫来一位 coder 检查我的代码,之后进入会议室再次面谈,在交谈过程中,我感觉到这位面试官其实不懂技术,问的问题基本都是项目经历以及碰到过哪些问题如何解决之类的。还问过我之前面试过哪些公司,我简单介绍了一下,以及说了一下期望的薪水。然后面试官说也可以给到我差不多的薪水,但是由于本人不是太想去这家公司,所以就委婉的拒绝了。

面试难度: ★ ☆

评价: 其实面试官不懂技术,很好忽悠,但是机试会考验一个人的能力,好在机试不算难,所以比较好过。

工作环境: ★ ★

评价: 公司整体给人的感觉还可以。

## 东方通

这家公司是我近期面试中,面试时间最久的一家公司了,从下午 1 点一直面试到下午 5 点多,

这家公司主要是做中间件,也就是所谓的 web 容器。一开始先是笔试,不过笔试题倒是给人眼前一亮的感觉,并不是从网上拷贝下来的那些千篇一律的问题,而且经过用心准备,考验应聘者对 java 的理解程度。最难的应该就是那些代码补全的题目,如果没有看过 LinkedList、HashMap 这些类的源码,还真是很难回答上来,我答的也是踉踉跄跄。由于我面试的部门是做中间件维护工作的,所以到了面试的时候,面试官第一个问题就是了解过哪些 web 容器,因为我也就看过 tomcat 的源码,而且看的也不全面,所以只能说 tomcat 了。然后就开始是问我 tomcat 的实现原理以及 request、response、session 这些东西是如何实现的,我把自己对 tomcat 的理解基本上都叙述了出来,面试官觉得还可以,就继续问我 ArrayList、LinkedList、HashMap、ConcurrentHashMap、HashSet 这些类的实现原理以及多线程跟线程池的原理。最后开始问 jvm 的问题,第一个问的就是 jvm 的体系结构,接着问类的加载过程,方法区、堆、栈分别各存放哪些数据。然后就顺理成章的开始给我介绍该部门的工作内容与职责,由于本人想做 j2ee 的开发,不希望做中间件的维护工作。所以就跟面试官说了一下情况,面试官却说 J2EE 开发部门目前不缺人,让我考虑一下这边的工作。由于本人态度比较坚决,不希望做中间件的维护工作,面试官说让我等一下,他去找 J2EE 部门的领导来跟我聊一聊。当时我心里暗想: "你妹啊,你不是说 J2EE 开发部门不缺人嘛! "



## 近期面试经历总结及公司评价(下) II

因为当时 J2EE 部门的领导在开会,所以等了大概将近 20 分钟,才开始第二轮面试。问的问题也不是很难,基本上都是一些 Struts、Spring、Hibernate 方面的问题以及之前的项目经验。之后就是让我稍微等一会儿,过了一会儿,人力进来说 BOSS 有事不在,要等半个多小时才能回来,之前的那套题是中间件部门的,现在我面试的是 J2EE 开发部门,让我重新做一套题。当时我心里极度不爽,怎么还要做题啊。。结果发现人力不是给我的笔试题,而是直接把笔试题的答案给了我!真是天助我也,我就什么也没干的一个人在会议室待了半个多小时,之后 BOSS 进来问我题做完了吗?我说:”人力直接把答案给我了。“,BOSS 说:”那你怎么不跟他说啊“,我说:”我已经知道答案了,再做也没什么意义,而且这套题跟第一次我做的那套题差不多。“。之后就开始问我职业规划方面的问题,我也大概说了一下。最后人力找我谈工资的事情,我要的薪水该公司给不到,所以最后就此作罢。

面试难度: ★★★☆

评价: 由于之前我面试的部门是做中间件,所以对 JAVA 基础要求很高,之后的 J2EE 部门面试难度不是太大。

工作环境: ★★★

评价: 作为一家民营企业,公司有茶水间会提供免费的茶水以及盲人按摩,还有每月的活动经费,个人觉得已经很不错了。

豆丁网

豆丁网的笔试题足足有 6 页,实在是让人无

语,幸好后面大多都是叙述题,基本上都十个字以内搞定。6 页的题,本人半个小时就搞定了,之后到了会议室进行面试,面试官是两个人,这两位让我联想到了三国杀里面的吕布跟马超。因为问的问题实在太难,每问一个问题,都仿佛是在向我出“杀”,而且一个需要出两张“闪”,另一个如果判断成功则此“杀”不可闪避,直接就掉血。下面我就说一下那几个让我掉血的问题吧。问题一:“java 中引用又分为哪几种类型? 垃圾回收器是如何处理每种类型的?”,好吧,此“杀”需要两张闪,所以本人实在不知道,选择直接掉血。问题二:“HashMap 的实现原理是什么? 如果不使用链表,还有什么其他实现方式?”,HashMap 的实现原理倒是“闪”了过去,结果第二个问题不清楚,结果又掉血。问题三:“一个拥有 2 亿用户的表,你会如何设计它,让它达到更好的性能?”,此杀判定成功,不可闪避,继续掉血。。其实后面还陆陆续续的问了很多问题,由于时间过的比较久,已经记不起来了。因为问的问题比较难,我回答的也一般,面试官就说如果薪水低一些是否可以接受,我说了一个底线,低于这个就不会考虑了,他们说回去讨论一下,有消息会通知我。到迄今为止都没有消息,估计是被 pass 掉了。

面试难度: ★★★

评价: 问的问题对我来说实在太难,再加上本人也没有大数据量处理方面的经验,所以回答的不是很好。

(本文未完,请阅读原文)■

<http://developer.51cto.com/art/201209/355221.htm>

# 面试时，如何向公司提问？

今天，我看到硅谷招聘经理 Steve Buckley 的一篇文章，正好探讨了同一件事的另一面：应聘者如何向公司提问。

很多人将面试看作一种单向选择，事实上，面试是一种双向选择：不仅是公司挑选你，也是你挑选公司。面试就是为双方提供互相了解的机会，公司在评估你，你也在评估公司。

面试官也知道这一点，所以他们有心理准备，期待你提出问题，并且会做出回答。所以，面试时不要浪费向公司提问的机会。而且，你主动提问，表明你比较重视这个职位，会加深面试官对你的印象，可能会提高面试的成功率。



## 有一些注意点，你需要知道

1. 面试之前，一定要做准备，多了解公司的情况。
2. 你提出的问题，应该围绕“这份工作是否合适我”这个中心点，其他与应聘关系不大的问题，不宜多问。
3. 提问的时候，要自然放松，不要害羞，就把它当作普通的聊天。你要表现出对公司的真诚兴趣。
4. 提问要直接了当，不要绕圈子。提出问题

之后，你要保持安静，让面试官多说话。

5. 面试官回答的时候，你可以做笔记，或者事先询问能不能做。笔记必须简短，你的大部分时间，要用来全神贯注倾听面试官的回答，并与其有眼神的交流。

6. 面试结束后一周内，最好打一个电话或发一封邮件，了解公司对你的反馈意见。即使面试失败，你不妨也问一下原因，这会有助于你以后的面试。

下面是一些你可以问的典型问题。

## 问题一：你们为什么要招聘这个职位？

Q1: Why are you currently recruiting for this position?

这个问题会使得面试官开始谈论当前的项目，或者谈论前一位离职人员。无论哪种情况，都会让你了解，一些与你最密切相关的公司情况。

## 问题二：你们的新员工多吗？

Q2: Do you have many new staffs?

这个问题起一个过渡作用，使得谈话导向公司内部的情况。但是，它本身也能说明一些问题。如果公司成立已经超过四年，又没有新项目，但是新员工却很多，这往往说明公司文化不是很健康。

## 问题三：你们公司（团队）目前面临的最大挑战是什么？

Q3: What are the biggest challenges your team are facing right now?

（本文未完，请阅读原文）■

<http://developer.51cto.com/art/201208/354098.htm>

## 一淘曲琳：搜索引擎的架构维护与职责协调

编者按：51CTO 记者在 International ArchSummit 架构师峰会上，针对一淘的搜索引擎技术对一淘网架构师曲琳的采访，谈谈一些搜索引擎的技术与研发的协调与合作关系。

编者按：51CTO 记者在 International ArchSummit 架构师峰会上，针对一淘的搜索引擎技术对一淘网架构师曲琳的采访，谈谈一些搜索引擎的技术与研发的协调与合作关系。



曲琳，一淘网架构师，主要在一淘网负责搜索引擎技术。

**51CTO 记者：**首先，能简单的介绍一下自己么？

我是 08 年浙大毕业的博士生，毕业之后我就去了 alibaba，当时是在 B2B 搜索中心在做 alibaba 中文站和国际站的搜索后台，之后又到了阿里云，又到了淘宝，而现在是在一淘。而整个过程里都是在搜索这个领域上，垂直搜索这一块做引擎开发工作。大概的背景就是这样。

1) 你大学是学什么专业的？

我不是计算机专业的，我是学电子信息技术及仪器，应该是属于近计算机专业吧。我博士研究方向是挖掘相关的，是智能监控方向。其实具

体在学校做的事情是嵌入系统这些，偏嵌入系统的软件。当然我们这个方向是有硬件有软件的，所以这个背景会对硬件更加的了解一些。

2) 当时学的东西和你后来做的搜索你觉得关系大吗？

你如果说领域的话其实是相差蛮大的，当然也略有些联系，在相关性方面，其实都会用到一些数据挖掘方法。但是在引擎平台上讲具体的领域其实是有很大大偏差的，当然在这个软件设计或者是开发这个角度其实它的基础背景都是差不多的。刚开始来 Alibaba 的时候是有些挑战的，一个全新的领域需重头学习。

3) 您一开始加入 Alibaba 的时候，是有资深的工程师带您么？

Alibaba 这个文化，就是你去这个公司会有个师兄，然后他会在不仅仅是你工作上的，包括你生活上有一有困难的时候都可以跟他商量。我们叫他 Buddy 也好，更多的是一种平等的方式，你可以直接交流。他并不是你的直接领导，你的主管可能是另外一个人。这个 Buddy 可能是会稍微在公司呆过一段时间的，对公司比较熟悉了解的，这样一个相对比较资深的人。在这个过程当中来带领你，所以你的工作在他的帮助下一点点地开展起来。

## 一淘曲琳：搜索引擎的架构维护与职责协调 II

51CTO 记者：一淘网的搜索技术部主要承担了哪些职责？

现在在一淘的搜索技术部其实是承担除了阿里云之外的整个 Alibaba 集团的搜索业务，都在一淘搜索，包括淘宝的主搜索，还有一淘本身的搜索，还有 Alibaba B2B 的中文站、国际站、速卖通，所有搜索后台都是在一淘搜索这个部门，都是由我们来支持的。

1) 那往其他部门部署也是由你们来支持吗？

其实并不是这样的，这个我们可能会说在那边的业务，在一个具体应用上的业务就是由我们这边的人来负责的，并不是那边部门由具体的人来负责我们给他提供代码不是这样的。我们直接是提供服务的，包括引擎的运维，从开发到最终上线，到业务需求的满足都是在一淘搜索的。

2) 相当于他们是你们的客户，他们说我们要搜索服务，然后你们提供给他们？

我们是以服务化形式来支持各个部门，支持各应用业务。

51CTO 记者：工程师们跟业务部门是如何协作的？

一般来说直接跟工程师打交道的是产品部门，当然也会有些运营部门来提需求，我们针对每个需求方都会有专门的技术的接口，那可能在具体的开发过程当中会有个项目经理这样的角色，可能是从技术团队当中出这么个角色来协调这个整个的开发过程，还有需求的沟通都会有这样的一个角色来承担，来协调。那么我们可能对每一个具体的业务建议会有专门的团队来支持，比如说我，其实是负责一淘的搜索后台，那么我们会另外的一些小团队负责淘宝主搜索的，针对每一

个具体的应用都会这个业务需求的定制，还会有这种针对业务的性能优化，还有引擎的部署，调优，甚至会有些非搜索的这个服务。其实搜索引擎并不仅仅是这个核而已，你为了提供完整的服务，可能还会有些 key-value 的东西，或者是有一些在线计算实时计算的一些服务，还会有一些离线的，数据挖掘的，算法相关的，还可能会有些比如索引 build，数据的抓取，或者是 dump，这个一整套的工作都会有针对业务的定制。

4、那其实像你刚刚说这个协调产品和技术双方的角色是不是对两边都非常熟悉？

对。一般来说你要协调这个项目的話你首先要有非常多的这个项目经验，另外一方面你得对整个技术的上下游，其实这个搜索的战线是蛮长的，从后到前经历的各个角色很多，首先你得有这个经验你得了解整条线的这个情况，这样的话你才能在各技术方案当中选择一个比较合理的。这里可能还不涉及到架构的问题，在做具体的业务需求更多的是怎么样把这个流程看起来能够选择一个方案来实现需求。另一方面你也得了解业务，不仅仅要了解技术，可能你要从具体的项目当中了解整个产品的一个规划，甚至对用户的体验你要有感觉，怎么样做才是更合理的方向，或者在这个需求的 tradeoff 时候你得找到一个比较合适的度，不能一味的满足这个产品的想法，跟这个技术实现的难度当中要做些折中的，这里面其实对项目接口那个角色的要求是蛮高的。

1) 一般在 Alibaba 做到这样的能力需要大概 3-5 年？ ■

本文更详细内容，请查看：

<http://developer.51cto.com/art/201208/354633.htm>



# 基于Java开发的开源QQ客户端iQQ

iQQ 使用 Java 语言跨平台开发,基于腾讯 WebQQ 3.0 网络协议。可以使用于 Java 所支持的各种平台上运行。作者基于 Linux(Ubuntu 12.04) 系统,使用 IDE NetBeans 开发,以源代码公开方式贡献给各个程序爱好者,希望可以一起开发此项目。



iQQ 使用 Java 语言跨平台开发,基于腾讯 WebQQ 3.0 网络协议。可以使用于 Java 所支持的各种平台上运行。作者基于 Linux(Ubuntu 12.04) 系统,使用 IDE NetBeans 开发,以源代码公开方式贡献给各个程序爱好者,希望可以一起开发此项目。

iQQ 使用 Java 语言开发,运行时需要安装 Java 运行环境。

如果系统已经安装好 Java 运行时,直接点击 run.sh 方可运行 iQQ

或者 iQQ.jar 右键使用 Java Runtime 进行运行

Java 运行时安装方法:

1、Debian, Ubuntu 等系统。

在命令行中,键入:

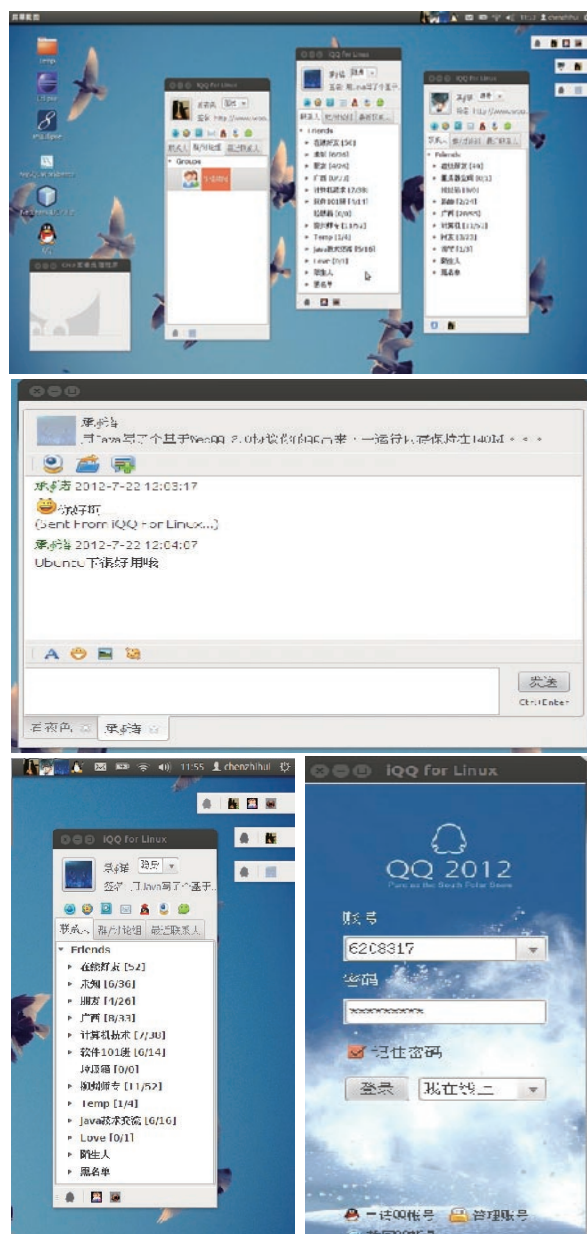
```
sudo apt-get install openjdk-6-jre
```

或者 Ubuntu 下直接在软件中心直接搜索 jre 选择 OpenJDK java 6 运行时

2、Fedora, OracleLinux, Red Hat Enterprise Linux 等系统。

在命令行中,键入:

```
su -c "yum install java-1.7.0-openjdk"
```



iQQ 下载地址: <http://code.google.com/p/iqq/downloads/list>

# 招聘优秀的开发人员：简历是很危险的

我到目前为止已经面试了很多工程师了。我越来越感觉到我对于招聘的最初的那些想法都是错的,例如简历。以前,在面试之前,我总是会仔细的阅读简历。我会查看候选人的过去经历,他们的技能还有兴趣爱好。但是随着时间的推移,我发现,简历是很危险的。



我到目前为止已经面试了很多工程师了。我越来越感觉到我对于招聘的最初的那些想法都是错的,例如简历。以前,在面试之前,我总是会仔细的阅读简历。我会查看候选人的过去经历,他们的技能还有兴趣爱好。但是随着时间的推移,我发现,简历是很危险的。

我经常在看简历的时候想“这哥哥很牛,看看他的经历,明显他精通 Ruby 和 JavaScript,面试也许就是走个形式”。但是当真正面试的时候,我发现这哥哥几乎不会写代码,分不清变量的作用域。最终的结论是,简历和面试结果关系不大。

简历最多可以让你知道该问候选人什么问题。但是简历也会让你误以为你很了解候选人,包括他的能力。

确实,简历是很多公司解决大量招聘的途径。但是我认为针对工程师的招聘,我们可以通过代码的形式来解决。代码能够过滤掉很多不合适的人员。让工程师面试工程师也可以提供很有用的信息。

我现在看简历的第一件事情就是 GitHub 链

接。如果我能找到 GitHub 链接,我就不再继续看简历了。我会开始阅读代码。虽然开源的代码不代表全部,但起码是个切入点。我花五分钟的时间阅读候选人的代码,我就对这个人的技术水平还有编码习惯有了一个大致的了解。在面试的时候会顺利很多。

当我和候选人交谈的时候,我们会聊他的代码。我会问一些问题,有关他们为什么要这么写代码,背后的原因和逻辑。整个面试都是在讨论代码和编写代码。这才是软件工程师每天的工作,这才是重要的内容。

我想要雇佣一个我愿意共事的人。我想要雇佣比我优秀的人,我可以向他们学习的人。

招聘是一个公司最基本的工作。Valve 的招聘手册:

把招聘工作做好是世界上最重要的事情。没有任何其他事情的重要性可以和它接近。它甚至比呼吸还重要。下面是一些我在评估候选人的时候会问的问题;

- 我可以接受这个人成为我的老板吗?
- 我可以从这个人身上学到很多吗?
- 假如他为竞争对手工作会怎样?

我的经验是评估候选人,而不是简历。在面试过程中注重那些解决常见问题的代码。还有,雇佣比你优秀的人。■

## ■ 编者按

多年来我一直想学 VBA——微软 Visual Basic 的改造版、用来编写 excel 宏的编程语言,但由于各种原因一直拖延着没学。它并不是一种很难学的语言,只是因为它不是我的主要工作内容,我一直没腾出时间来学它。

## 编程将死, 程序员永存

多年来我一直想学 VBA——微软 Visual Basic 的改造版、用来编写 excel 宏的编程语言,但由于各种原因一直拖延着没学。它并不是一种很难学的语言,只是因为它不是我的主要工作内容,我一直没腾出时间来学它。昨晚,我跟随 Twitter 上的一个链接发现了 DataNitro,五分钟后我学会了 VBA。

DataNitro 是个 excel 插件,能够让你在 Excel 里运行 Python 脚本,因为我在某种程度上算是擅长 Python,我根本不需要使用 VBA。也许这并不是在 5 分钟内学会了 VBA,而是成功的完全绕过了它。对于我来说,这跟学会了它一样有效。

大家是否还记得在《骇客帝国》里的那个 DVD 播放器? 它能连接到你的大脑来教会你功夫。DataNitro 就像那个东西,只是它是给 Excel 笨蛋准备的。它很可能帮我节省了数百小时在 VBA 里面挣扎、自学、尝试、纠错的时间。



骇客帝国 学功夫

我写这篇文章不是来为 DataNitro 做广告的,而是想谈谈对编程这种工作发展方向的思考。

几天前, Brooklyn Bridge Ventures 的 Charlie O'Donnell 写了一篇非常好的博客叫做“最后一个程序员”,文章里他预测前端开发平台终有一天会致使整个软件开发部门废弃。当这些平台不断的增多不断的传播后,强大的软件开发需要的编码工作越来越少,越来越多的人能够参与进来。

### 程序员因此会消失吗?

这要看你如何定义程序员。如果这样说,那些每天花 15 小时在黑屋子里堆砌 0 和 1 的人才算是程序员,那我支持 Charlie 的论断,这种职业最终会被一点点的侵蚀掉,因为越来越优秀的“前端”平台会排挤掉众多的编码工程。

但从另外一个视角来看,我是一个“程序员”,因为这些平台能使我更快的开发出软件,而且不需要经过正规的培训,我想程序员的数量在很长的一段时间里都会不断的增加。

曾经有个时期,即使在 MD-DOS 上开发一个很简单的程序也要经过相当漫长陡峭的学习曲线过程。这越来越平坦的学习曲线归功于 Apple 和 Windows 开放的个人电脑走向大众。类似的,当各种平台使开发工作越来越容易后,越来越多的人能轻松的上手编程,而不需要花 3 或 6 个月的时间去学 VBA。■

本文更详细内容,请查看:

<http://developer.51cto.com/art/201208/354950.htm>

## 美程序员12岁少年开发98款游戏

北京时间 8 月 30 日消息,据国外媒体报道,在培养高质量工程师方面,美国正在逐渐被印度和中国甩到后面,因此美国正在小学和中学阶段普及计算机编程课程。与此同时,以 Lua 为代表的入门级编程语言则成为美国青少年未来接触高级编程工具的敲门砖。

北京时间 8 月 30 日消息,据国外媒体报道,在培养高质量工程师方面,美国正在逐渐被印度和中国甩到后面,因此美国正在小学和中学阶段普及计算机编程课程。与此同时,以 Lua 为代表的入门级编程语言则成为美国青少年未来接触高级编程工具的敲门砖。

### 以下是文章内容全文:

年仅 12 岁的美国少年亚历克斯·福伊特 (Alex Foyt) 已经是在线游戏开发领域的老手了,他在 6 年时间里开发了 98 款游戏,其中包括一款躲避从天而降的胡萝卜和小鸡的生存挑战游戏。

福伊特掌握游戏开发的秘诀是:他此前学习了如何使用 Lua 语言进行编程的方法,而正是这种编程语言使他掌握了更加高级的软件开发工具。

居住在美国纽约州奥尔巴尼市的福伊特表示:“我特别希望能够成为一名计算机程序员,并且靠编程生活。”他最近长途跋涉到加州的圣克拉拉参加由游戏网站 Roblox Corp. 赞助的一个会议。Roblox 为用户提供了一个通过僵尸或中世纪堡垒等互动动画开发和操作游戏的平台

在网络游戏和以苹果 iPhone 为代表的设备程序大行其道的今天, Lua 已经成为帮助刚刚涉足软件编程领域青少年的入门视觉编程语言之一。包括 Roblox 在内的很多网站都在大力宣传

这些视觉编程语言。谷歌 (微博) 和思科经常抱怨美国工程人才出现断层现象,而这些编程语言也对解决人才短缺问题具有指导意义。

在美国斯坦福大学教授计算机科学的史蒂夫·库珀 (Steve Cooper) 指出:“工程人才短缺是美国科技市场增速放缓的一大主因。如果你去大学招聘会看看,就会发现雇主都扯着嗓子喊‘如果你是计算机科学家,快到我们这里来!’”

### 人才培养差距明显

在培养高质量工程师方面,美国正在逐渐被印度和中国甩到后面,因此让美国青少年早点接触编程十分重要。

经济合作与发展组织 (Organization for Economic Cooperation and Development, 简称 OECD) 2011 年针对在 25 岁到 34 岁之间就业的发达国家大学本科毕业生进行了一项调查,结果显示在获得过理科或工程学学位的毕业生比例排名中,美国仅仅列在全球第 23 位。

美国信息技术咨询公司 Cognizant Technology Solutions Corp. (CTSH) 总裁戈登·科伯恩 (Gordon Coburn) 指出:“这种差距在小学阶段就体现出来了。到他们上大学时,他们根本没有掌握数学知识。因此市场紧缺高质量的编程人员,只要我们发现类似人才,有多少我



## 美程序员 12 岁少年开发 98 款游戏 II

们招多少。”

虽然 Lua 编程语言已经有 20 年的历史,但直到最近几年才又重新焕发出了生机,这还要归功于 Roblox。这家网站成立于 2005 年,截至今年 7 月,该网站的独立访问用户已达 1,100 万,用户年龄多数都在 8 到 14 岁之间。而且,Roblox 用户已经累计开发了超过 800 万款游戏。

### 从入门到高级

移动应用程序开发平台 Corona Labs Inc. 也在使用 Lua 编程语言。而在与另外两大入门级编程语言 Alice 和 Scratch 的竞争中, Lua 也表现出了积极的上升势头。

美国计算机科学教师协会( Computer Science Teachers Association, CSTA )主管克里斯 史蒂芬森( Chris Stephenson )指出:“在过去几年里,我们已经看到吸引学生加入编程队伍并教授他们基本编程知识的趋势呈现爆炸性增长。Alice 和 Scratch 两种编程语言非常流行,因为学生喜欢它们,而且能够利用这些语言开发出真正有创意的程序。”

居住在犹他州西班牙福克市( Spanish Fork )的罗伯特 内伊( Robert Nay )在他 14 岁的时候就使用 Corona 的社交开发软件套装开发出了一款名为《泡泡球》( Bubble Ball )的游戏。这是一款利用 iPhone 的位置感应器晃动手机而让小球绕过障碍到达特定位置的游戏,内伊完全是利用课余时间,自学编程、设计开发了这款游戏。这款游戏在 2010 年 12 月 29 日上架销售,至今的累计下载量超过 1,500 万。在苹果 App Store 免费应用下载排行榜上,《泡泡球》在数周内一度超过了全球最热门的游戏《愤怒的小鸟》。

内伊在接受采访时表示:“我只是开发了一款我自己想玩的游戏而已,我认为这款游戏很有趣。对于未来的事业选择,我倾向于从事与计算机有关的行业。”

在掌握了入门级编程语言后,内伊还学习了更加高级的 Java 编程语言用来建设网站以及开发游戏和程序。

### 教授解决问题技巧

美国杜克大学计算机科学学院院长布鲁克 奥斯伯恩( Brook Osborne )表示,基于 Alice、Scratch 和 Lua 编程语言的程序开发平台有助于在小学和中学普及编程知识,教会这些学生解决问题的技巧,并将学生的编程兴趣拓展至包括 Python、Java、Ruby 和 C++ 等其他编程语言。

奥斯伯恩指出:“当你理解了编程的概念并掌握了程序员的思维结构以后,学习计算机句法就没有任何障碍了。”

一些年轻的程序员正在通过在线编程课程学习编程技术,包括 Udacity Inc.、Codecademy 和 Coursera Inc. 在内的初创企业就提供这样的在线课程。Codecademy 的联合创始人扎克 西姆斯( Zach Sims )指出,自从 2011 年 8 月推出在线编程课程以来, Codecademy 的用户已经超过 100 万,而且无论是小学教师还是大学教授都在课堂上使用该公司提供的编程资料。西姆斯还指出,学习编程正在成为一种“新素养”。

### 适合孩子的编程语言

史蒂芬森表示,在线编程课程和适合

本文更详细内容,请查看:

<http://developer.51cto.com/art/201208/355031.htm>

# Facebook引发的HTML5危机

@AppCan 刘鑫

近期几个新闻堆叠在一起,颇有韵味。先是 WHATWG 和 W3C 在 HTML5 标准上分道扬镳,继而“Facebook 移动应用宣布放弃 HTML5 的部分,改为纯 Native 方式开发”,接着又传闻苹果 AppStore 肃杀基于 Web 技术的 App。这几个事件对移动互联网行业来说个个都是重磅炸弹,押注 HTML5 的受到不小的打击,唱衰 HTML5 发展的借此幸灾乐祸。HTML5 真的只是一场政治斗争吗?到底 Facebook 为什么放弃 HTML5? 现阶段 HTML5 到底出了什么问题?

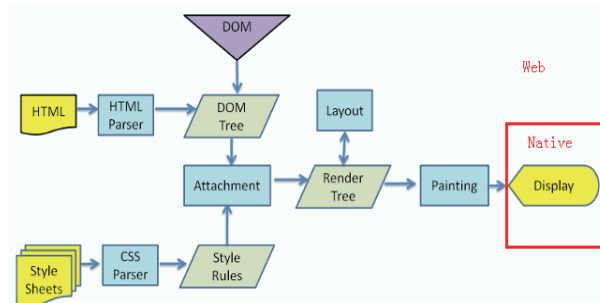
## Facebook 放弃 HTML5 主因:慢

“对于 Facebook 的 iOS 原生应用来说,它主要在三个方面有很大的速度提升:应用启动、共享新闻滚动还有图片点击查看。其总体速度大约提升了一倍。这个版本部分采用了 Facebook Camera 和 Facebook Messenger 两款应用的代码库:其中图片点击查看功能的代码是从 Facebook Camera 移植过来,而屏幕消息是从 Facebook Messenger 那克隆过来的。这个原生版本是由一个独立的团队开发,产品经理 Johnson 表示未来会充分利用公司的代码共享,也会适当向其他团队寻求帮助。”

上述摘自 Facebook 的官方博客。博客中介绍到 Facebook 的 iOS 原生应用放弃 HTML5 后速度得到大幅度提升。大家不禁好奇,为什么 HTML5 会比原生 NativeApp 要“慢”很多?

在当前的移动终端设备硬件配置和操作系统优化水平的前提下,大部分基于 HTML5 开发

的 Web 页面会出现延时加载展示的现象,也就是俗称的卡、慢。特别是在不同的视图界面 (view) 切换之间,这种卡和不流畅的现象会尤为严重。而 Native 应用不会出现这种情况。究其根源,在于浏览器解析的运作机制和原生 Native 的界面展示机制差异上。如下图所示:



红色框起来的部分是原生 NativeApp 的界面展示机制,简单的看起来就是 1 个步骤——展示,因为所有的绘图和渲染工作都由系统直接完成。而红框以外的部分包括红框内的部分是 webkit 核心的浏览器解析页面的流程。相比 Native 的 1 个步骤,webkit 的解析过程可谓漫长而艰辛。历经解析、建立 Dom 树、获取对应资源、布局、建立渲染树、绘图到展示。所以不管移动终端设备硬件如何发展,这个差异是始终存在的,最多只是随着硬件的提升和软件的优化将这个差异收缩到最小甚至忽略。

更糟糕的是。Facebook 之前的 iOS 混合了 HTML5 的移动应用,使用 HTML5 绘图的页面在 HTML5 开发上也毫无技巧可言,基本沿用了主流前端开发框架 jQuery mobile 等的单 View 多 div 的机制。也就是在一个网页内绘制多个视图,页面之间的切换其实只是一个页面内不同区块的

## Facebook 引发的 HTML5 危机 II

切换。这种方式加大了浏览器的渲染和绘制工作强度。并且在数据加载和流量上产生很大的负面影响。如果切换到新页面,之前的页面不进行销毁,则会加大运算量和增加内存占有,而如果销毁又会导致已经下载的数据失效,要重新载入,浪费流量。类似情况在中国的网络和设备情况下会尤为突出。所以 Facebook 不当的在 Native App 内混搭 HTML5 也难免引来用户怨言。

还有,一如报道中提到的, Facebook 这次的改进提升主要是“新闻滚动和图片点击”。如果了解 HTML5 的人,就会发现,这两点当然是“不应该在现阶段使用 HTML5 实现的”。为什么?笔者作为一个基于 HTML5 技术的 Hybrid App 系统的设计者,设计秉承的一个原则就是“凡是需要‘动’的部分和需要大量运算的部分,就最好使用原生弥补,而不是一定要使用 HTML5 来实现”。新闻滚动,这种不停通过改变 Dom 树进而改变渲染再绘图展示的使用场景相比原生 Native 弱势是非常明显的。至于图片的部分就更不用多说了,这并不是 HTML5 眼下擅长的部分。HTML5 现在擅长的部分是数据量不大的页面、动画少的页面,特别是跨平台的开发。充分利用好 HTML5 的优势,尽量降低 HTML5 的弱势,学会用好 HTML5,才是现在这个时期使用 HTML5 开发的重点。可以说开发技巧很重要。

### 现阶段 HTML5 的问题: 政治斗争

“原生版本是一个独立团队开发的。” Facebook 公开的这一点也耐人寻味。原来客户端是 Native 与 HTML5 混合的方式,原来的团队也肯定有原生的开发能力,为什么非要一个独立

团队重新耗费 6 个月进行重新开发?或许这里不能排除公司内政治因素,而 HTML5 成为一个牺牲品。HTML5 的政治不仅是一个公司内的,更是整个行业的。7 月份,同为 HTML5 制定者的 WHATWG 和 W3C 表示无法继续合作,前者希望制定一个能够跟随市场或技术动态的标准;后者则要确立一个“死”的标准,一旦正式颁布再也无法修改。



WHATWG 和 W3C 的分道扬镳或许会成为 HTML5 发展的一个分水岭。WHATWG 背后有 Google、苹果, W3C 拉到了特立独行的巨无霸微软。标准是为利益服务的,曾经力推■

本文更详细内容,请查看:

<http://developer.51cto.com/art/201208/355089.htm>

# 我在外包公司的一年测试生涯

作者 / 熊志男

2012年8月1日,到公司整整一年,前些天提出了加薪申请。接下来思考下一年的工作得失。

## 初印象

多数时候,我是有些自卑的,一部分源于自己的不自信,另一部分源于对公司的没信心。人们说起谁谁在“百度”“腾讯”,一副羡慕嫉妒恨的样子,哈喇子都流出半截。如果说起某某外包公司,禁不住浑身打个哆嗦,嘴角都歪到下巴去了。我不知道国外的外包企业都是这样,还是只有在中国如此狼狈。

可想而知,告诉别人我在外包公司工作,就好像一个人做了什么龌龊的事情。我一直在想,人要面对真实的自己对吧,如果不能勇于面对自己对于职业的选择,就简直是个十足的懦夫了。

为了证明以上观点不是我自己独创的,我拿出两个现实中的例子:

我组里的同事,在公司呆了5年多了,每每提起这个,她会生气,生气也许是同事间假装的矫情。但是没有感觉到一丁点荣光。

传说中的一位已经离职的前同事,北大毕业高材生,从公司离开后去另一家优秀企业应聘,考官疑惑不解“你怎么还在##公司呆过?”潜台词是你是不是能力有问题,或者简历有造假嫌疑,为什么不在更好的公司?

## 不一样

来公司之前,和大多数人的想法一样,打死也不进外包公司。苦、累、没归属感等等名词已经印在我脑子里。可是当我实在找不出理由继续呆

在上家公司,而现在的公司又抛出热情的橄榄枝的时候。面对换工作压力的我,如释重负。当时只有一个念头“即使是火海,也跳了!”

庆幸的是,我找到了几个不一样的地方:

不外派,ODC(offshore delivery center)项目,那就是在公司内部办公。那么即便是小,也有自己的组织,我没有被外派过,无法体会外派人员的心情。只是道听途说,外派会没有归属感,低人一等。

和谐快乐的团队氛围,也许别人看起来一般,我却感觉很快乐。因为快乐不快乐,只有身在其中才能体会到。也许团队还是那个团队,只是我的心态发生了变化,我开始明白,只有敞开心扉融入集体,才能体会到集体的快乐。

精明强干的女领导,好吧,如果说是拍马屁,我也是发自内心的。而且不是指某一个女领导,而是其中几个,如果对号入座,应该有那么两三个。我的直系领导交给我负责自动化测试的任务,慢热型的我磨叽好几天才搞明白框架。她上来噼里啪啦就完成了Web Service client测试框架。佩服一个人不因为他职位比你高,而是因为他能做到的你不能轻易做到。

我们隔壁组的已经离职的“王总”,爽朗的北京女孩,带给我们快乐的北戴河旅程,组里比较木纳的我和小王都玩High了。隔壁组的准妈妈Leader,口头禅是“不会玩的孩子不是聪明■

本文更详细内容,请查看:

<http://developer.51cto.com/art/201208/354734.htm>